(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 January 2001 (11.01.2001)

PCT

(10) International Publication Number
WO 01/03011 A2

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/US00/18443

(22) International Filing Date: 30 June 2000 (30.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/141,993 1 July 1999 (01.07.1999) US
60/185,602 28 February 2000 (28.02.2000) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier applications:
US 60/141,993 (CIP)
Filed on 1 July 1999 (01.07.1999)
US 60/185,602 (CIP)
Filed on 28 February 2000 (28.02.2000)

(71) Applicant *(for all designated States except US)*: NETMORF, INC. [US/US]; 655 Boylston Street, Boston, MA 02116 (US).

(72) Inventors; and

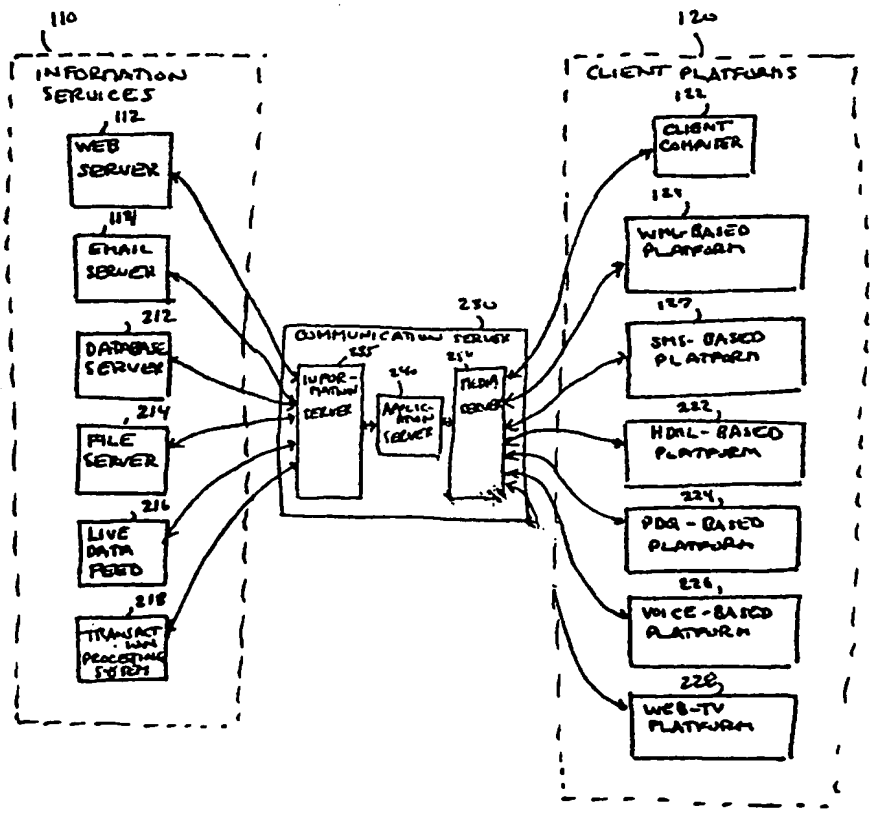
(75) Inventors/Applicants *(for US only)*: KAMADOLLI, Shyam [IN/US]; 1775 Massachusetts Avenue #6, Cambridge, MA 02140 (US). TIPNIS, Rajeev [IN/US]; 139 Webster Street, Needham, MA 02496 (US).

(74) Agent: WALPERT, Gary, A.; Fish & Richardson, P.C., 225 Franklin Street, Boston, MA 02110-2804 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR,

[Continued on next page]

(54) Title: CROSS-MEDIA INFORMATION SERVER



[Continued on next page]

WO 01/03011 A2

HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(57) **Abstract:** A system and method for coupling diverse client platforms and diverse information services without necessarily requiring separately configured modules to couple each combination of client platform and information service. Content from any of the information services is first converted into a common format and then translated into a format suitable for the target platform. The approach is applied to client platforms supporting wireless communication to personal devices such as cellular telephones and personal digital assistants, and to information services such as Internet-based servers, messaging servers, transactional systems, and live data feeds.

CROSS-MEDIA INFORMATION SERVER

Background

This invention relates to access to information over a variety of communication media.

A variety of different communication infrastructures are available today over which users of personal client devices can access information from information services, such as from Internet-based server computers. These personal devices include, for example, cellular telephones, personal digital assistants (PDAs), and pagers. These devices use a variety of communication protocols to access data from the information sources. For example, some cellular telephones support data communication using the Wireless Application Protocol (WAP) and Wireless Markup Language (WML) to provide access to text-based information from the cellular telephones. Some handheld devices, such as the Palm VII computer made by Palm Inc., make use of a mobile radio communication infrastructure (SMR) and use a Palm Query Application (PQA) protocol to provide information access from the handheld device. Other devices use various other application protocols, such as the Handheld Device Markup Language (HDML) or specific versions of the eXtensible Markup Language (XML), to control how information is accessed and displayed, and use various communication protocols are used to pass information between the devices and information services. In the description that follows, the communication infrastructure for passing data to and from a personal device and the personal devices that make use of that infrastructure are referred to jointly as a client platform over which data is accessed.

Information services that are accessed from the personal devices generally use protocols that depend on the nature of the information services. For example, an electronic mail server may provide access to email using the Post Office Protocol (POP) application protocol using the Transport Control Protocol/Internet Protocol (TCP/IP) over the public Internet, while a World-Wide-Web (WWW) server may provide information using the Hyper-text Markup Language (HTML) and hyper-text transport protocol (http) using TCP/IP.

Referring to FIG. **1**, an example of a prior approach to supporting various information services **110** and various client platforms **120** makes use of content translators **130** to couple the data sources to the client platforms. In the example shown in FIG. **1**, information services **110** include a Web server **112** and an Email server **114**. Client platforms **120** include a client computer **122**, which executes software for interacting directly with both Web server **112** and Email server **114**. Client platforms **120** also includes a WML-based platform **124** and an SMS-based platform **127**. In this example, WML-based platform **124** includes a WAP infrastructure **127** for communicating with a WML device **128**, such as a cellular telephone, over a wireless

communication system over which data is passed using the WAP protocol. Only a single representative WML device **128** is shown in the figure, although it is understood that a client platform generally supports a large number of client devices. SMS-based platform **127** includes a paging infrastructure **128** and a representative SMS device **128**. Content translators **130** include separate translation modules for coupling each of the information services **110** with the different of the client platforms **120** that do not directly support the protocols or formats of the information services. In this example, a Web-WML module **132** converts HTML-based content into WML based content while a separate Web-SMS module **134** converts HTML-based content to an SMS-format. One approach to this conversion is to configure the modules with data associated with particular Web pages that have been selected to be accessible from the client devices, and this data identifies the portions of the Web pages that should be translated into the target formats. For example, graphic components and long text sections may be ignored and only specific content may be extracted for translation into the target format. This approach to translating HTML pages is often referred to as "scraping" the pages to extract the desired content. If the layout of an HTML page changes, both the Web-WML and the Web-SMS modules may have to be reconfigured to accept the changed layout. Content translators **130** also include an email-WML module **136** and an email-SMS module **138** that provide translation services between email server **114** and WML-based platform **124** and SMS-based platform **127**, respectively.

Summary

In a general aspect, the invention provides a system and method for coupling diverse client platforms and diverse information services without necessarily requiring separately configured modules to couple each combination of client platform and information service. Content from any of the information services is first converted into a common format and then translated into a format suitable for the target platform.

In one aspect, in general, the invention is a method for communicating between information services and client devices. The method includes accepting a first request for information from a first of the client devices, which includes receiving a first identifier which identifies requested information. The method then includes forming a message encoding the requested information. Forming this message includes (a) accessing a specification of the requested information based on the first identifier, (b) identifying a first of the information services from the accessed specification, (c) accessing the information service to retrieve a part of the requested from said information service, and (d) formatting the requested information according to a common format that is independent of the client device from which the first request was accepted. The method then includes passing the requested information to the client

device, including translating the message from the common format into a format that is associated with the client device and that is different than the common format, and transmitting the translated message to the client device.

The method can include one or more of the following features:

Accepting the first request, forming a message, and passing the requested information are repeated for requests from a number of the client devices, and passing the requested information to the client devices includes translating messages from the common format into a number of different formats each associated with different of the client devices.

The common format identifies layout elements, and the plurality of different formats associated with the client devices includes one or more markup formats.

The method further includes determining one or more parameter values associated with capabilities of the client device, and forming the message includes encoding the requested information using the parameter values to determine the content of the message.

Translating the message from the common format includes applying a translation procedure that is independent of the requested information.

Translating the message from the common format into the format associated with the client device includes translating from a first markup language to a second markup language.

The second markup language is WML, HDML, HTML, compact HTML, PQA, generic XML, or Avant.Go.

In another aspect, in general, the invention is a system for communicating between a number of information services and a number of client platforms. The system includes an application server, which includes an interface to the information services, a storage coupled to the application server for holding configuration data, and a media server for communicating messages with the client platforms according to a number of message formats. The media server includes a number of modules each associated with a different one of the client platforms and including an interface to the client platform for communicating with the client platform according to one of said message formats. The system includes a communication path coupling the application server and the media server for passing information between the information services and the client platforms in a common format that is independent of which of the information services and client platforms are involved in the communication.

The system can include one or more of the following features:

Each module of the media server includes a translator for converting messages from the common format to the message format for communicating with the client platform associated with said module.

The system further includes an information server coupled between the application server and the information services. The information server includes a number of modules, each

associated with a different one of the information services. A communication path couples the information server and the application server for passing information in a format that depends on a class of the information service involved in the communication. The information server thereby provides a generic interface for the application server for classes of the information services.

The system also includes a configuration application for providing the configuration data based on input from an operator of the system.

The system also includes a cache storage for holding information received from the information services.

The invention includes on or more of the following advantages:

Multiple different types of client devices, each using a different message format, can be supported without having to specifically configure translators for each combination of information service and type of client device. Also, unlike many "scraping" approaches to format translation, translation from the common format to the format for a client device does not necessarily require configuration of the translator to be able to extract desired portions of particular messages, such as particular web pages, based on their location in the messages.

In versions of the system in which parameters associated with the capabilities of a client device are available while forming the message in the common format, that message can then be directly translated into the format for the client device without having to perform significant processing to match the message content with the capabilities of the client device. This simplifies the design of the format translators.

The invention enables easy addition of new information services or client platforms. For example, addition of a client platform does not have to address mapping multiple information services to that client platform since only mapping of the common format to that platform is required. Similarly, adding of modifying an information service does not require updating configurations for multiple client platforms.

Other features and advantages of the invention are apparent from the following description, and from the claims.

Description of Drawings

FIG. **1** is a block diagram showing multiple information services coupled to multiple client platforms through content translators;

FIG. **2** is a block diagram showing multiple information services coupled to multiple client platforms through a media server according to the present invention;

FIG. **3** is a block diagram of a communication server;

FIG. **4** is a flowchart that illustrates processing of a client request; and

FIG. 5 is a block diagram that illustrates a cascaded arrangement of communication servers.

Description

Referring to FIG. 2, according to this invention, a communication server **230** provides communication and data translation services to couple information services **110**, which includes a number of separate information services, and client platforms **120**, which includes a number of separate client platforms, each of which typically supports a large number of personal client devices. Communication server **230** is a programmable computer, or alternatively a system that includes a number of interconnected computers. A storage device **210**, such as a magnetic disk, holds software that when executed on the communication server implements a number of interrelated software modules that provide the communication and data conversion services that couple the information services and the client platforms. Communication server **230** has communication links to information services **110** over a variety of communication channels, such as over the public Internet, over a private data network, or over a secure or dedicated communication link. Communication server **230** also has a number of communication links to client platforms **120**.

Communication server **230** hosts three principal modules: an information server **235**, which communicates with the various information services **110**; a media server **250**, which communicates with the various client platforms **120**; and an application server **240**, which passes information between the information server and the media server. The separate information services of information services **110** are grouped into a number of generic classes, such as database servers, file servers, messaging servers, and marked-up information servers. Communication between information server **235** and application server **240** does not necessarily depend on a particular information service which is involved in the communication, but rather, in general, depends on the generic class of the information service. Communication between application server **240** and media server **250** is largely independent of the particular information service or its generic class, and is independent of the particular client platform involved in the communication. In this embodiment, information is passed from application server **240** to media server **250** formatted ("marked-up") according to a specific instance of the eXtensible Markup Language (XML) in such a way that media server **250** does not, in general, have to deal with specific characteristics of the information services. Media server **250** provides information to application server **240** regarding the capabilities of the client platform, such as the size of the graphical display on a client device or a maximum data size that can be transferred over the communication infrastructure of the client platform to the client device, and application server **240** provides information to media server **250** in a format that the media server can process

without having specific knowledge regarding a content-specific format or layout of the information.

In FIG. 2, a variety of separate information services of information services **110** are illustrated. Information server **235** is configured to interact with each of these services using interfaces that are native to those services and to interact with application server **240** according to the generic classes of those services. Information services **110** include Web server **112** and Email server **114**, as shown in FIG. 1. Information services **110** also include a database server **212**, such as an Oracle or Sybase SQL server, and a file server **214**, for example, providing an interface using the File Transfer Protocol (FTP) over the Internet, or a software module which provides an interface to a local file system. Information services **110** also includes a live data feed **216**, such as a stock ticker, and a transaction processing system **218**, such as a banking system. In alternative embodiments, not all these information services are necessarily present, and additional information services and generic classes of services may also be included.

A variety of client platforms are also illustrated in FIG. 2. In addition to client computer **122**, WML-based platform **124**, and SMS-based platform **127**, which were introduced in FIG. 1, client platforms **120** include an HDML-based platform **220**, such as a cellular telephone platform in which HDML based browsers are supported. In some embodiments, a single cellular telephone infrastructure supports multiple types of devices, such as WML and HDML based telephones, and communication server **230** supports both protocols through a common communication path. Client platforms **120** also include a PDQ-based platform, which supports communication with PDAs, or alternatively a PQA, C-HTML, or Web Clipping Application based platform. Client platforms **120** also optionally includes a voice-based platform **226**. In this platform, the client device is a voice telephone and the communication infrastructure includes a speech recognition server. The speech recognition server is configured to provide access to information. The speech recognition server uses the VoXML markup language that specifies a spoken dialog using a text-based specification. Client platforms **120** also optionally includes a Web-TV platform **228**. In this platform, the client device is a set-top coupled to a television receiver. In alternative embodiments, not all these client platforms are necessarily present, and additional information services may also be included.

Referring to FIG. 3, information server **235** includes a number of information modules **335**, each associated with a different one of the separate information services of information services **110**. An information module **335** interacts with a separate information service using the native interface for that information service. For instance, in the case of database server **212** (shown in FIG. 2), the associated information module **335** makes native database queries to that database server and accepts responses using that native format. In another instance, in the case of email server **114** (see FIG. 2), the associated information module **335** uses a standard Internet

protocols such as IMAP, POP, or SMTP. Each information module **335** interacts with application server **240** through an information interface **342** using a protocol that, in general, depends on the generic class of the information service associated with that information module. For instance, in this embodiment in the case of database information modules, application server **240** sends text-based commands or queries to the information module, and receives in return two-dimensional text-based tables of responses. In another instance, in the case of an information module **335** for a marked-up information service, application server **230** sends text-based requests to the information module and receives nested tagged text-based content, in this embodiment following an XML-based syntax.

Media server **250** includes a number of media modules **350**, each associated with a different separate client platform of client platforms **120** (FIG. **2**). Each media module **350** communicates with application server **240** over a media interface **344** using a protocol that is independent of the particular client platform associated with that media module, and communicates with its associated separate client platform according to the requirements of that client platform. The requirements of a particular platform include both the format of information that it receives from communication server **230**, and data channel requirements. With regard to data channel requirements, some media modules **350** communicate with their associated client platforms over the Internet, while others may use leased telecommunication lines, or other communication links. Some client platforms may have a maximum message or packet size for communication, which is different from that of other client platforms. With regard to data format requirements, each media module **350** formats information it receives from application server **240** into a specific format required by its associated client platform, and possibly a specific format required by the destination personal device being serviced on that platform. For example, one media module **350** may handle WML-based requests from a cellular telephone platform, while another module may handle PQA-based requests from a PDA platform. A client platform that supports WML-based telephones may support a variety of models of telephones, each with slightly different capabilities, such as the number of lines or pixels of display, and media module **350**, in general, formats the information based on the capabilities of that personal device. Media server **250** also includes an additional module, trigger module **355**, which is used by information services or other external systems to initiate asynchronous events that are not specifically initiated from the client devices, for example to push notification data to client devices. Trigger module **355** communicates with application server **240** through a trigger interface **346**.

In this embodiment, data modules **335**, media modules **350**, and trigger module **355** are implemented as procedures that are invoked by application server **240**. That is, information interface **342**, media interface **344**, and trigger interface **346** include application-programming

interfaces (APIs) that define how information is passed between the application server and those modules. Some of these procedures are implemented as dynamically-loadable libraries (DLLs), allowing additional modules to be added without necessarily changing the remaining portions of the system. In alternative embodiments, other implementation approaches are used, such as a distributed implementation in which communication between the application server and the information and media modules passes over a data network.

Application server **240** makes use of configuration data **310** that provides information for use by the application server to process requests received from client platforms **120** and to access particular of the information services **110**. The range of information, or "content," that application server **240** can access from information services **110** can be viewed as a single "address space" of content. When a client device sends a request to access information through communication server **230**, it sends an identifier of that information. The identifier can be viewed as an address in the address space that is handled by application server **240**, which is hosted on the communication server. In this embodiment, these addresses take the form of text specifications of locations of information in a hierarchical naming scheme, in an approach that is similar to that used for Uniform Resource Locators (URLs) for Web-based content. Within this hierarchical naming scheme, configuration data **310** includes a number of view specifications **315**, each of which has a distinct name in the naming scheme. In this embodiment view specifications **315** use an XML-based syntax, which is described more fully below, that allows application server **240** to identify and communicate with particular information services to retrieve the information requested from the client device. Using configuration data **310**, application server **240** aggregates content from the various information services and provides a uniform interface to the client devices. Communication server **230** also hosts a configuration application **320** that supports an operator or administrator of the system in specifying configuration data **310**. Alternatively, configuration application **320** can be hosted on another computer that communicates with the communication server.

Each request from a client platform, which is passed by the associated media module **335** to application server **240**, is mapped by application server **240** to one of the view specifications **315**. A view specification **315** acts somewhat as a template for forming a response that application server **240** passes to media server **250**. In this embodiment, the view specification also includes procedural and conditional elements, and instructions related to how particular information services should be accessed.

A view specification uses an XML markup syntax, which includes nested fields each identified by a tag. A set of predefined tags define an abstraction of a layout, or more generally a structure, of a response to a request as well as define methods of obtaining content for the response from the information services. A *PAGE* tag defines the top level of a view

specification. Note that although the view specification refers to a "page," when presented on a client device, the presentation may span multiple screens. An example would be a page that includes a menu of choices, where the number of choices is greater than can be displayed at one time on the display of the client device. Another example is a presentation that includes multiple portions, which are often referred to as "frames." A *PAGE* field optionally includes a parameter which is a hint to application server **240** whether these portions should be presented in sequence, or in "parallel" using a top-level menu to select the portions when presented on the client device.

A *PAGE* can optionally include a *TITLE* element, which is a text string that may be used as a heading for the view when presented on the client device, or when used for navigating, for example, in a history of views visited by the client device.

A *PAGE* can include any number of *TEXT*, *TABLE*, *FORM*, or *NAVBAR* elements, which define displayable elements of a view. A *TEXT* element includes a text specification, typically of a character sequence to display on the client device. A *TABLE* element represents data in columns, where the data can be statically defined in the view specification, or can be obtained from an information service. A *FORM* element is similar to an HTML *FORM*, and is used to retrieve input from a client device, and to display results on the client device in a row and column arrangement. A *NAVBAR* element is used for a "navigation bar," which is a kind of menu. A *NAVBAR* element includes a set of sub-elements, *NAVBARITEM* elements, each of which logically defines an action to be performed if the user of a client device selects that item from the navigation bar.

Within certain elements, including within *NAVBARITEM* and *FORM* elements, a *CALLBACK* element is used to specify data that is passed to a client platform and later may be passed back to application server **240**. In the case of a *CALLBACK* element in a navigation bar, the data is passed back when the user of the client device chooses the associated item in the navigation bar. The data that is passed back to the application server can identify another of the view specifications **315**, and optionally provides parameter values to the application server.

To support access to information services **110**, a *PAGE* can include one or more *QUEURYSOURCE* fields. A *QUERYSOURCE* field identifies a particular one of the separate information services. In this embodiment, the individual information services are numbered, and the *QUERYSOUCE* element identifies the source information service by its number. The *QUERYSOURCE* element also identifies the generic class of the information service, and includes one or more *QUERY* elements, which define specific requests to be made of the information service. For example, in the case of a database class of information service, the request might be one or more SQL statements that are used to access information in the database. In another example, in the case of an HTML-based information service, the request might be data that would be provided to a Web server in an http *POST* command. In the case of an XML-

based information service, the request might be a complete URL, optionally including both a pathname and variables to be used by a script executing at the information service. The *QUERY* element also includes an identifier that is used to reference the results of the query elsewhere on the page. In this way, a page can include multiple *QUERYSOURCE* elements that are used to access data in different information services, and each *QUERYSOURCE* element can include several *QUERY* elements, thereby allowing application server **240** to aggregate multiple pieces of information onto one page.

Various sub-elements of a *PAGE* can reference a result returned from an information source, identified by the name of the query that generated the result, as well as a specification of a portion of the result. In the case of a database information service, the reference to the portion o the result is by row and column. Also, in populating a table, the rows and columns may be computed based on the positions within the populated table. In a markup based information service, the reference to the portion of the result makes use of the nested tags in the result. In general, the form of the reference to the result depends on the structure of results associated with the generic class of the information service.

In addition to referencing content obtained from information services **110**, a view specification **315** can also reference variables passed from media server **250** to application server **240**. The values of these variables may come from the client devices, for instance through callback mechanisms. Another type of variable is generated at media module **350**. An example of such a variable is a "hint" that the media module passes to the application server to characterize the capabilities of the client device. These hint variables may identify the data format handled by the client device, and specific capabilities of the client device. For instance, the hint variable may identify that the client device accepts WML based data and that it has 4 lines of display. Application server **240**, based on a view specification **315**, makes use of these hint variable to generate information that is suitable for essentially direct translation to the format handled by the client device with requiring that media module **350** perform complex translation that depends on the specific content being passed to the client device.

View specification also includes elements that allow conditional processing of a request from a media module **350** based on data from an information service or on data provided as a hint by the media module. A *CONDITIONS* element within a page defines a set of one or more named or numbered Boolean expressions that the application server evaluates. Other elements on the page are conditionally inserted based on the value of the named or numbered Boolean expressions.

Application server **240** creates a formatted page in response to a request from a media module **350** using the referenced view specification **315** and content it receives from information services **110** through information modules **335**, and passes this formatted page to the media

module. The formatted data uses an XML-based specification in which the elements generally correspond to the displayable elements of the view specification. That is, the XML that is generated can include elements that correspond to text blocks, tables, forms, navigation bars, and these elements can include callbacks that are used for navigation. Application server #0240 tailors the XML it generates to the specific characteristics and requirements of the client platform using the hints passed to it from the media module.

Referring to the flowchart shown in FIG. **4**, a request for information from a client device is handled in several steps. First, a user enters a request on the client device, for example, using an embedded browser application on the device to make a selection from a menu of possible sources of information (step **410**). For instance, in a cellular telephone, an embedded "micro" browser application may be stored in ROM. This browser application makes use of a limited-capability display on the phone to allow the user to navigate among the types of information available to the user. Note that each of the information services **110** are not necessarily distinguishable to the user since media server **230** performs an aggregation of those services and translates particular requests from a client device to access appropriate of the information services.

The request is then transmitted over the communication infrastructure associated with the client device (step **412**). In the case of a wireless device such as a cellular telephone, the request is transmitted over a wireless communication infrastructure to a gateway server that provides a link between the wireless infrastructure and a wired data network infrastructure that is external to the client platform. Depending on the client platform, this gateway server may reformat the request to conform to a particular application protocol (step **414**). For example, the gateway server may translate between a communication protocol supported by embedded micro browser in a client cellular telephone and a standard application protocol such as WML or HDML. In other client platforms in which the client device itself supports a standard protocol, such as HTML, the gateway server may simply provide a data routing function to pass data between the wired and wireless communication infrastructures.

The request is then transmitted from the client platform to communication server **230** (step **416**). In this embodiment, the requests from various client platforms are transmitted to communication server **230** over the Internet. In alternative embodiments, other communication links, such as dedicated communication links to particular client platforms are used. Communication server **230** receives the request and passes it to media server **250**, which is hosted on the communication server. Within media server **250**, the media module **350** that is associated with the type of client device that made the request handles the request.

Media module **350** interprets the request according to the application format used by the client platform (step **418**). In particular, the media module determines an identifier of a view

specification **315** for generating the information being requested from the client device. Media module **350** sends this identifier to application server **240** (step **420**). In addition to sending the identifier, depending on the particular client platform, media module **350** optionally sends "hints" to application server **240** that are parameter values which relate various constraints and capabilities of the client platform. For instance, on an embedded browser of a cellular telephone, only a limited number or menu choices may be easily managed. Media module **350** can optionally send a hint that specifies the maximum number of lines that is requests to be provided from the application server module. These hints may depend on the particular client device making the request. For example, in a cellular telephone environment, different models of telephone may be supported, and these various models may have different display capabilities. To the extent that information related to which model of client device is involved, the hints sent by media module **350** to application server **240** may depend on the model of client device. The hints that the media module provides the application server may also depend on the nature of the communication infrastructure. For example, in a low-speed wireless environment, the media module may identify a maximum desired data size based on the data link speed.

Next, application server **240** receives the request from media module **350** of media server **250**. Note that the format of this request is generic in that it does not depend on the particular client platform involved in the communication. The format of the requested view, and the format of any hints provided by the media module to the application server are essentially independent of the client platform, other than as encoded by the hints supplied to the application server.

Application server **240** accesses a view specification **315** in configuration data **310** to process the request (step **424**). For each of queries in the view, application server **240** accesses the corresponding information service to retrieve data according to the data query component of the view element, and then formats the retrieved data according to the display template of the view element (step **426**). The specification of the view may include procedural components that are executed by the application server, for example to make use of hints that were provided by the media module.

Application server **240** then passes the formatted information to the media module **350** that made the request (step **428**). If a complete result is too large to send at one time to the client device, application server #0240 sends a partial result to the client device while making the remainder of the result accessible. Application server **240** includes a *CALLBACK* element, which is displayed to the user as a "more" choice for example. The *CALLBACK* element includes sufficient information for the application server to later generate the remainder of the result. For example, if a query returns more rows than can be displayed on a client device, the callback includes information that allows application server #0240 to later query the information

service to return the next "chunk" of data. Using this callback approach, application server **240** does not necessarily have to maintain a state of an interaction with a client device since the callback data encodes enough information to continue the interaction.

Media module **350** receives the formatted information, and translates the formatted information to a specific format for the client platform from which the request was received (step **430**). For example, the media module translates the common XML-based format used in this embodiment to a WML-based format supported by the client platform. Media module **350** may optionally perform other operations on the information. For example, in the communication infrastructure of the client platform limits the size of messages, the media module may split, or "chunk," the message into smaller pieces for transmission to the client device. Media server **250** sends the result to the gateway system at the client platform (step **432**) and the gateway system forward the result to the client device (step **434**).

In the description above, a client device requests information and that request is sent to the communication server and a response is sent back in a "pull" communication model. Certain client platforms also (or exclusively) support a "push" communication model in which information is sent from information services to client devices without being specifically requested. An example of such a system is a paging system, which may in fact not support any communication from the client device to the communication server. In this push model, an information service, or some other external system passes a signal to trigger module **355** in media server **250** (see FIG. **3**). The trigger module generates a request that it passes to application server **240** in much the same way that a media module **350** passes a request to the application server. For example, in the case of a notification that a particular user has received a message, an external messaging service would pass a signal to trigger module **355**. That signal identifies the destination client device, which directly o indirectly identifies the media module **350** and client platform in addition to the specific device, and identifies a view specification and optional parameters to request on behalf of the client device. Application server **240** accepts the request through its trigger interface **346** and processes the request in much the same way that it processes requests from media modules **350**. In particular, processing the request may result in accessing one or more information services through information server **235**. From the point of view of application sever **240** and information server **235**, the request is a "pull" from a client device, while from the point of view of the client platform, the resulting response is an unsolicited "push" of data. When application server **240** has created the formatted XML in response to the request, it passes the response to the appropriate media module **350**, which as identified using the data received from trigger module **355**, and the media module passes the request after reformatting to the client platform. In alternative embodiments, communication server **230** can include a timer that signals trigger module **355**, for example, periodically. These

signals can be used to poll information services and if necessary push content to particular client devices.

Referring to FIG. 5, communication servers can be arranged in a cascaded manner in which a communication server **230A** serves as one of the information services **110B** for a second communication server **230B**. Communication between communication server **230A** and communication server **230B** use the common XML-based format that is used to communicate internally between application server **240** and media server **250**. In this arrangement, the two levels of aggregation of information services may be performed. For instance, communication server **230B** may be administered by a particular company and used for access to its internal information services. In addition, general information that may not be specific to the company is provided through communication server **230A**, which, for example, is administered by a service provider and provided on a subscription basis to the company. The media module **350** at communication server **230A** that communicates with communication server **230B** performs a limited amount of processing on the formatted data it receives from the application server, with the majority of the translation into a device specific format being performed at communication server **230B**. Note that from the point of view of communication server **230B**, communication server **230A** is part of its information services **110B**, while from the point of view of communication server **230A**, communication server **230B** is part of its client platforms **120A**.

In alternative embodiments of the system, application server **240** includes a cache that it may use to avoid needing to perform queries through information modules **335** to the information services. In one such embodiment, the lifetime of a cache response is specified in a view specification **315**. That is, a *QUERY* element identifies whether the result should be cached, and also identifies the lifetime that the cached result may be reused. In this way, the information service itself does not necessarily have to identify a lifetime for the cached result, although in alternative embodiments, it may do so as well as or instead of specifying a lifetime in the view specification. In other alternative embodiments, media modules **350** may also include caches that allow them to reduce the number of requests that must be made of application server **240**.

The embodiments described above refer to a single communication server **230** that is coupled between information services **110** and client platforms **120**. As described above, communication server **230** is stateless. That is, communication server **230** does not necessarily keep information related to an interaction with a particular client device. In an alternative embodiment, the single communication server **230** is replaced with a bank of multiple essentially identical communication servers, and a load balancing mechanism passes requests from client platforms to selected communication servers. In order to improve performance of caches,

repeated requests from a particular client device or a particular client platform may be steered preferentially to the same communication server, although such steering is not essential.

In another alternative embodiment, communication server **230** uses a user management information service to store session-related information about individual users of client devices uses. This user management information service has several uses, including, holding server "cookies" and maintaining authentication information for users.

Certain information services operate under the assumption that a client of that information service can maintain a significant amount of state information, thereby allowing the information service itself to operate in a relatively stateless manner. For example, today, Web-based servers often rely on setting "cookie" data in a client to store information. A typical application of cookies is to store a shopping cart of items for an on-line merchant Web site. However, passing the cookie data back and forth to a client device may be prohibitive due to the amount of data that must be transferred, or the client may not have the capability to store cookie data or may have insufficient storage to store large amounts of cookie data. In this approach, cookie data is received by application server **240** from an information module **335**, and rather than passing the cookie data to the media module, it stores the data in the user management information service. The application service relies on a unique identifier of the client device to index the information it stores in the user management information service. In this way, application server **240** essentially acts as a proxy for the client device for processing cookies.

The user management information service is used similarly for authentication data. For instance, certain information services may require a user to be authenticated before requests for that user will be processed. In order to avoid having a user repeatedly have to re-authenticate himself, temporary credentials may be generated by an information service, or internally in the application service, and stored in the user information management service. Then, these temporary credentials are provided to the information service on subsequent requests from that user, until those credentials have expired. Expiration can, for example, be based on a total elapsed time or on an elapsed period of inactivity from the client device.

It is to be understood that the foregoing description is intended to illustrate and not to limit the scope of the invention, which is defined by the scope of the appended claims. Other embodiments are within the scope of the following claims.

What is claimed is:

1. A method for communicating between information services and client devices comprising:

accepting a first request for information from a first of the client devices, including receiving a first identifier which identifies requested information;

forming a message encoding the requested information, including

(a) accessing a specification of the requested information based on the first identifier,

(b) identifying a first of the information services from the accessed specification,

(c) accessing said information service to retrieve a part of the requested from said information service, and

(d) formatting the requested information according to a common format that is independent of the client devices from which the first request was accepted; and

passing the requested information to said client device, including translating the message from the common format into a format that is associated with said client device and that is different than the common format, and transmitting said translated message to said client device.

2. The method of claim 1 wherein accepting the first request, forming a message, and passing the requested information are repeated for requests from a plurality of the client devices, and wherein passing the requested information to the client devices includes translating messages from the common format into a plurality of different formats each associated with different of said client devices.

3. The method of claim 2 wherein the common format identifies layout elements, and the plurality of different formats associated with the client devices includes one or more markup formats.

4. The method of claim 1 further comprising determining one or more parameter values associated with capabilities of said client device, and wherein forming the message includes encoding the requested information includes using said parameter values to determine the content of said message.

5. The method of claim 4 wherein translating the message from the common format includes applying a translation procedure that is independent of the requested information.

6. The method of claim 1 wherein translating the message from the common format into the format associated with the client device includes translating from a first markup language to a second markup language.

7. The method of claim 6 wherein the second markup language is WML.

8. The method of claim 6 wherein the second markup language is HDML.

9. The method of claim 6 wherein the second markup language comprises XML, HTML, compact HTML, PQA, or Avant.Go.

10. Software stored on computer readable media comprising instructions for causing a computer system to perform functions of:

accepting a first request for information from a first of the client devices, including receiving a first identifier which identifies requested information;

forming a message encoding the requested information, including

(a) accessing a specification of the requested information based on the first identifier,

(b) identifying a first of the information services from the accessed specification,

(c) accessing said information service to retrieve a part of the requested from said information service, and

(d) formatting the requested information according to a common format that is independent of the client devices from which the first request was accepted; and

passing the requested information to said client device, including translating the message from the common format into a format that is associated with said client device and that is different than the common format, and transmitting said translated message to said client device.

11. A system for communicating between a plurality of information services and a plurality of client platforms comprising:

an application server, including an interface to the plurality of information services;

a storage coupled to the application server for holding configuration data;

a media server for communicating messages with the client platforms according to a plurality of message format, including a plurality of modules each associated with a different one of the plurality of client platforms and including an interface to said client platform for communicating with the said client platform according to one of said message formats; and

a communication path coupling the application server and the media server for passing information between the information services and the client platforms in a common format that is independent of which of the information services and client platforms are involved in the communication.

12. The system of claim of claim 11 wherein each module of the media server includes a translator for converting messages from the common format to the message format for communicating with the client platform associated with said module.

13. The system of claim of claim 11 further comprising:

an information server coupled between the application server and the information services, wherein the information server includes a plurality of modules, each associated with a different one of the information services; and

a communication path coupling the information server and the application server for passing information in a format that depends on a class of the information service involved in the communication,

whereby the information server provides a generic interface for the application server for classes of the information services.

14. The system of claim 11 further comprising

a configuration application for providing the configuration data based on input from an operator of the system.

15. The system of claim 11 wherein the application server further includes a cache storage for holding information received from the information services.

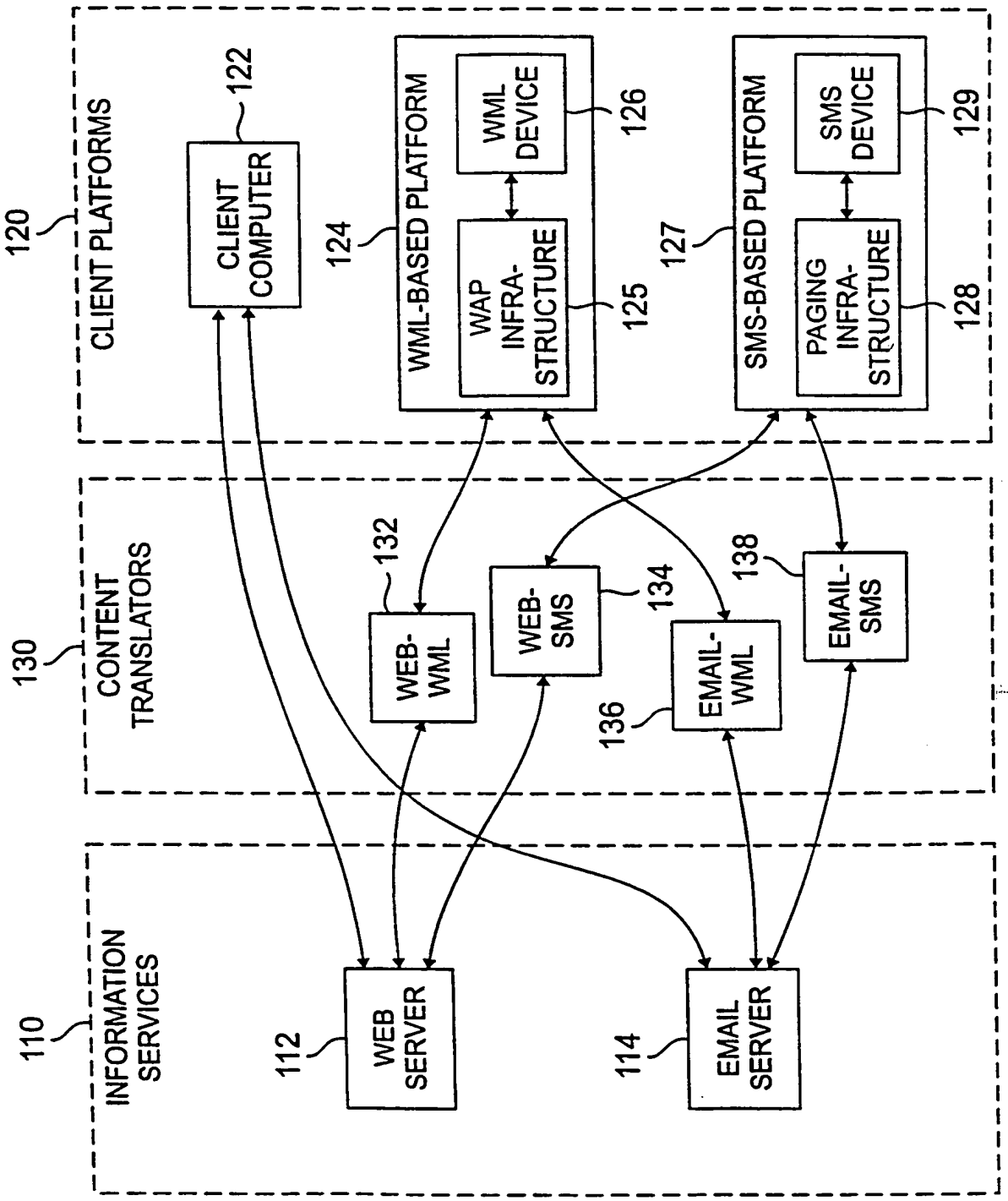


FIG. 1
(PRIOR ART)

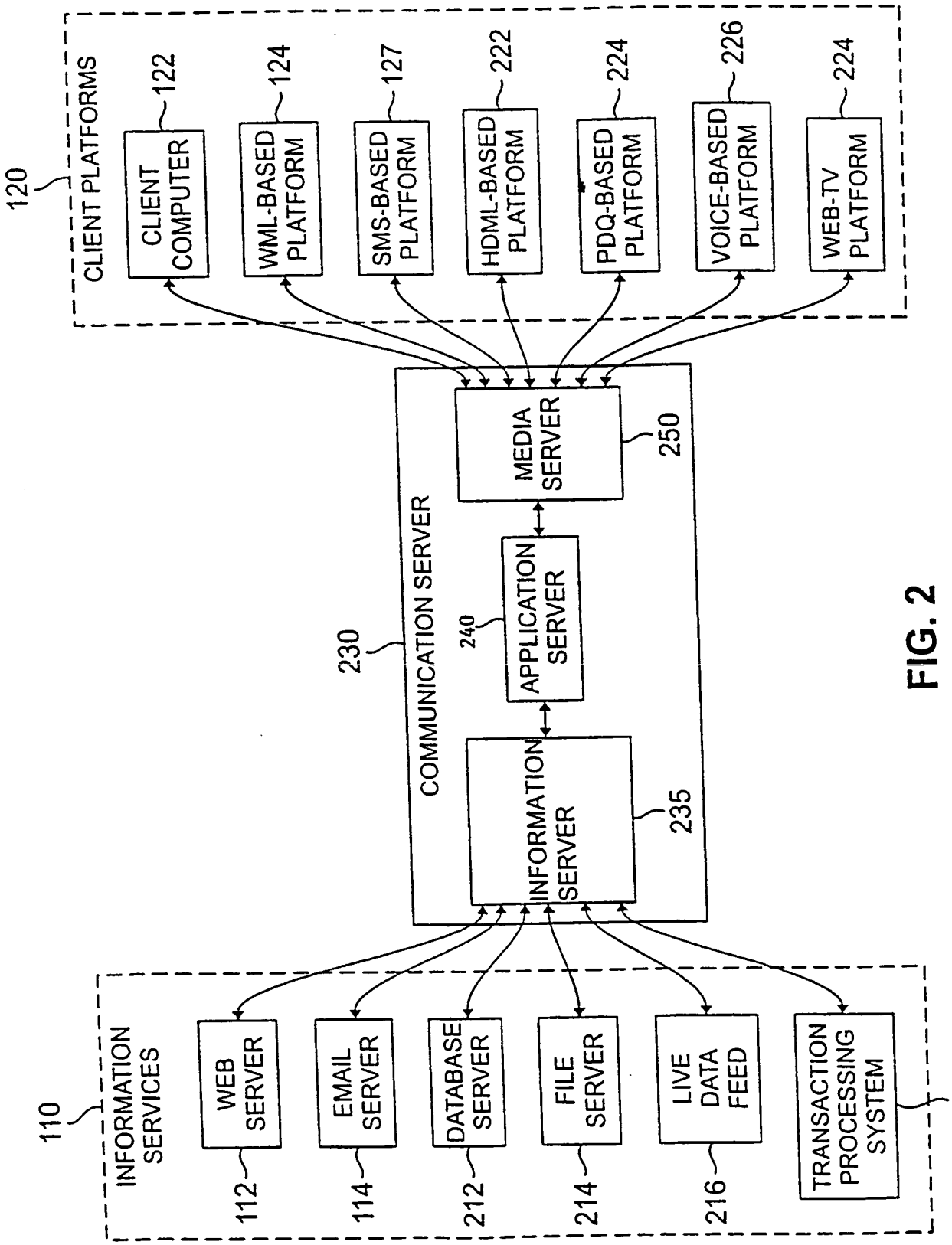

120
CLIENT PLATFORMS
CLIENT COMPUTER
122
WML-BASED PLATFORM
124
SMS-BASED PLATFORM
127
HDML-BASED PLATFORM
222
PDQ-BASED PLATFORM
224
VOICE-BASED PLATFORM
226
WEB-TV PLATFORM
224
230
COMMUNICATION SERVER
MEDIA SERVER
250
240
APPLICATION SERVER
INFORMATION SERVER
235
110
INFORMATION SERVICES
WEB SERVER
112
EMAIL SERVER
114
DATABASE SERVER
212
FILE SERVER
214
LIVE DATA FEED
216
TRANSACTION PROCESSING SYSTEM
218

FIG. 2

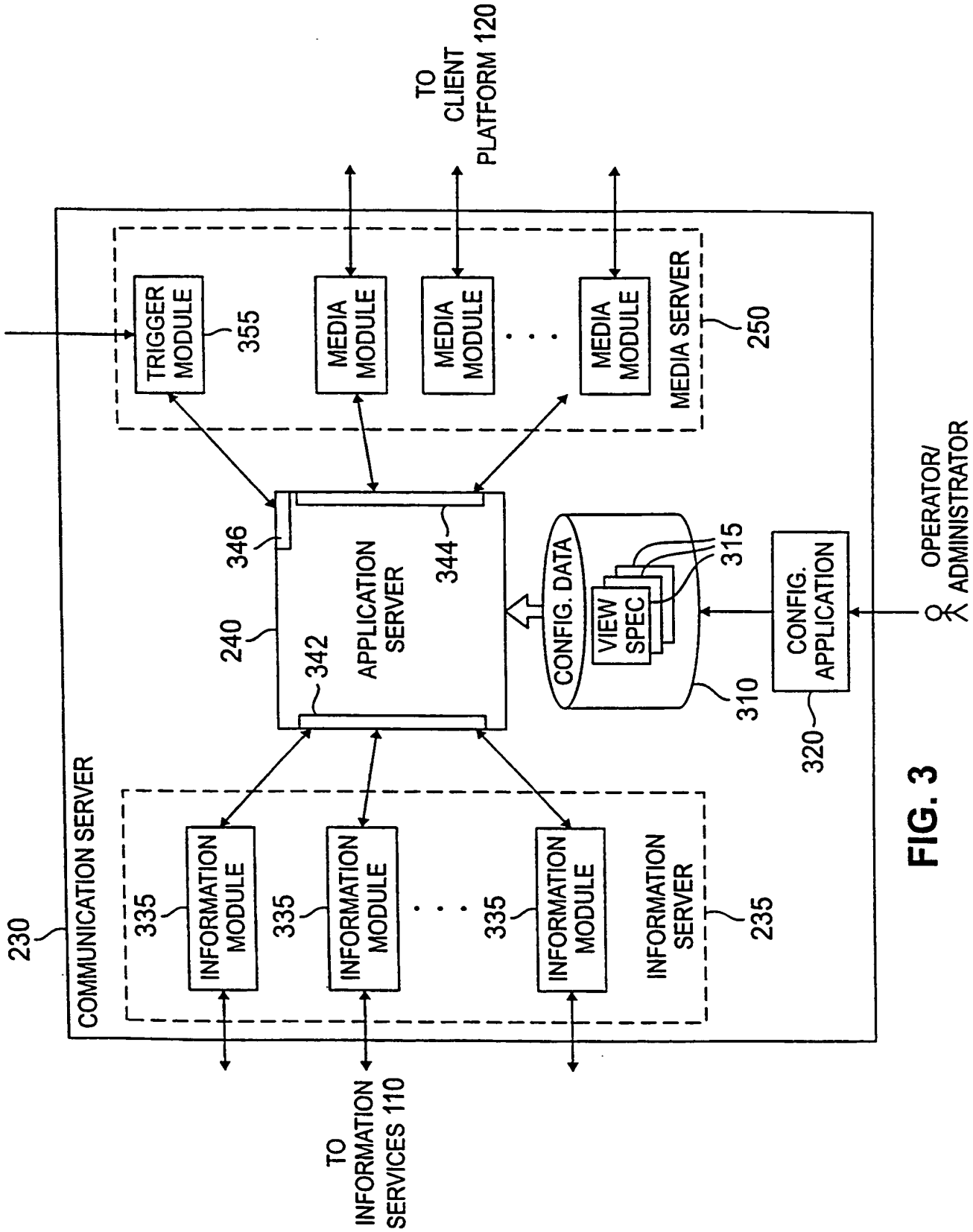

TO CLIENT PLATFORM 120
TRIGGER MODULE
355
MEDIA MODULE
MEDIA MODULE
MEDIA MODULE
MEDIA SERVER
250
346
344
APPLICATION SERVER
240
342
CONFIG. DATA
VIEW SPEC
315
310
CONFIG. APPLICATION
320
OPERATOR/ ADMINISTRATOR
230
COMMUNICATION SERVER
335
INFORMATION MODULE
335
INFORMATION MODULE
335
INFORMATION MODULE
INFORMATION SERVER
235
TO INFORMATION SERVICES 110

FIG. 3

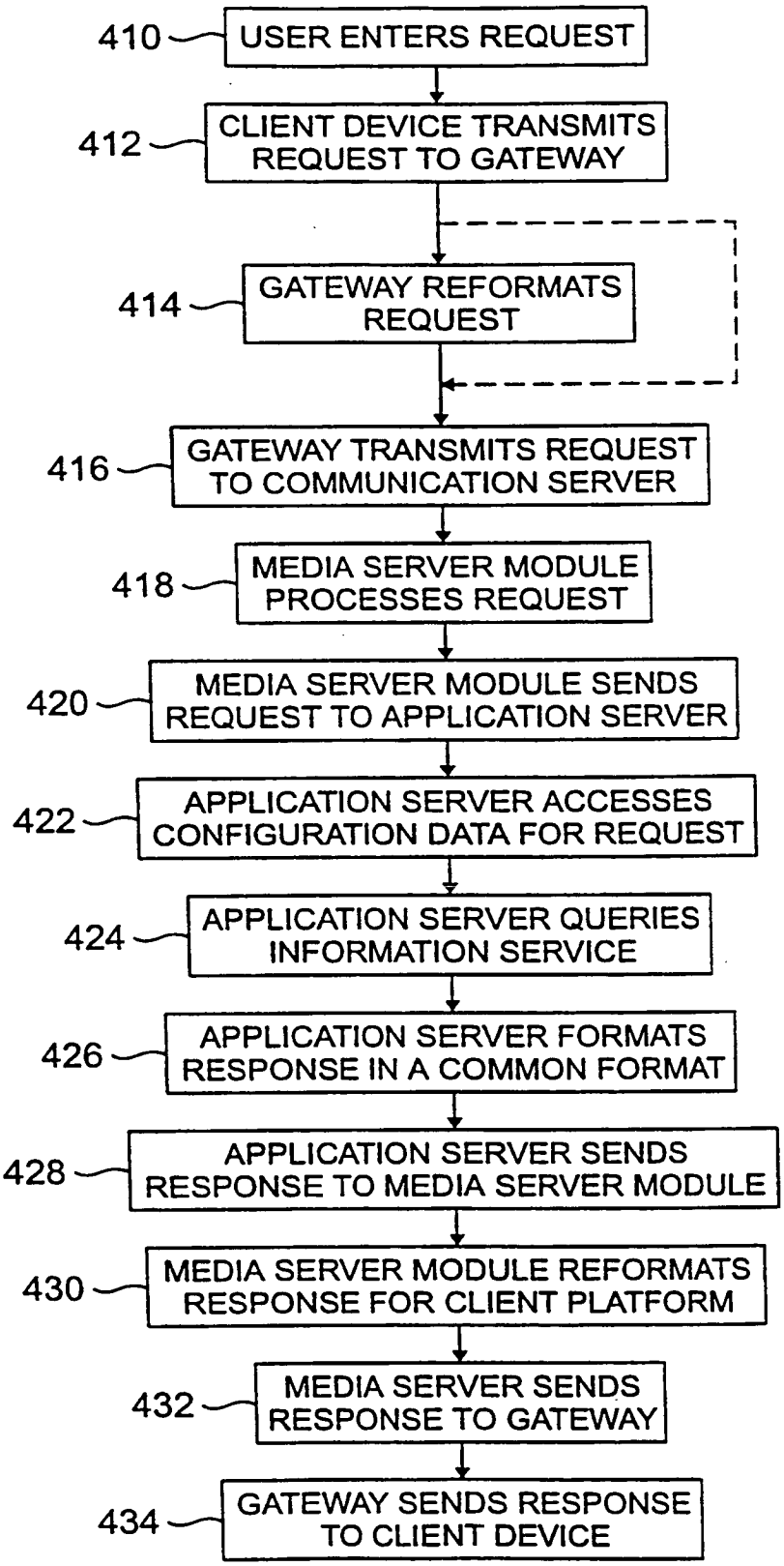

410
USER ENTERS REQUEST
412
CLIENT DEVICE TRANSMITS REQUEST TO GATEWAY
414
GATEWAY REFORMATS REQUEST
416
GATEWAY TRANSMITS REQUEST TO COMMUNICATION SERVER
418
MEDIA SERVER MODULE PROCESSES REQUEST
420
MEDIA SERVER MODULE SENDS REQUEST TO APPLICATION SERVER
422
APPLICATION SERVER ACCESSES CONFIGURATION DATA FOR REQUEST
424
APPLICATION SERVER QUERIES INFORMATION SERVICE
426
APPLICATION SERVER FORMATS RESPONSE IN A COMMON FORMAT
428
APPLICATION SERVER SENDS RESPONSE TO MEDIA SERVER MODULE
430
MEDIA SERVER MODULE REFORMATS RESPONSE FOR CLIENT PLATFORM
432
MEDIA SERVER SENDS RESPONSE TO GATEWAY
434
GATEWAY SENDS RESPONSE TO CLIENT DEVICE

FIG. 4

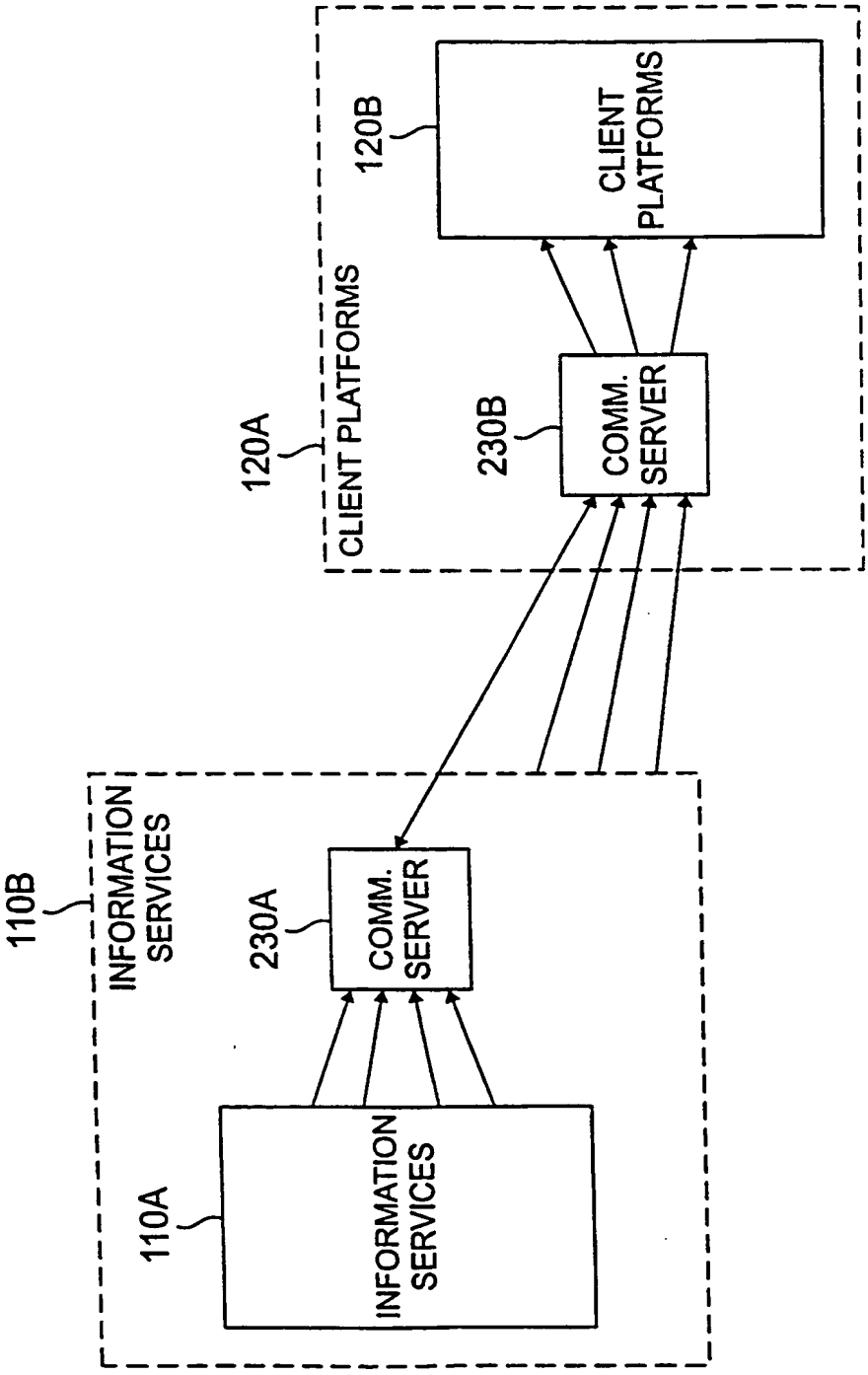

CLIENT PLATFORMS
120B
CLIENT PLATFORMS
120A
COMM. SERVER
230B
INFORMATION SERVICES
110B
COMM. SERVER
230A
INFORMATION SERVICES
110A

FIG. 5

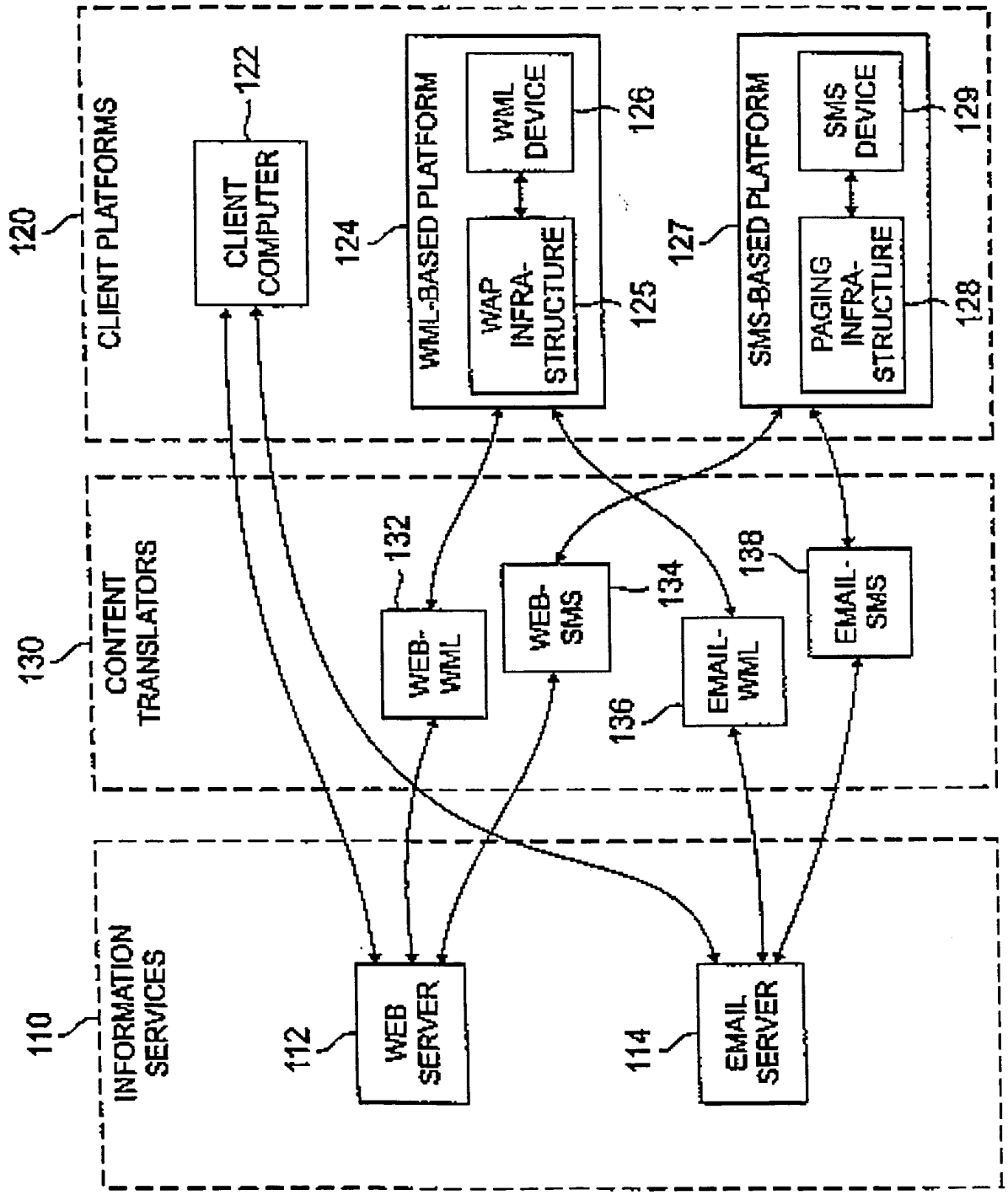


FIG. 1
(PRIOR ART)

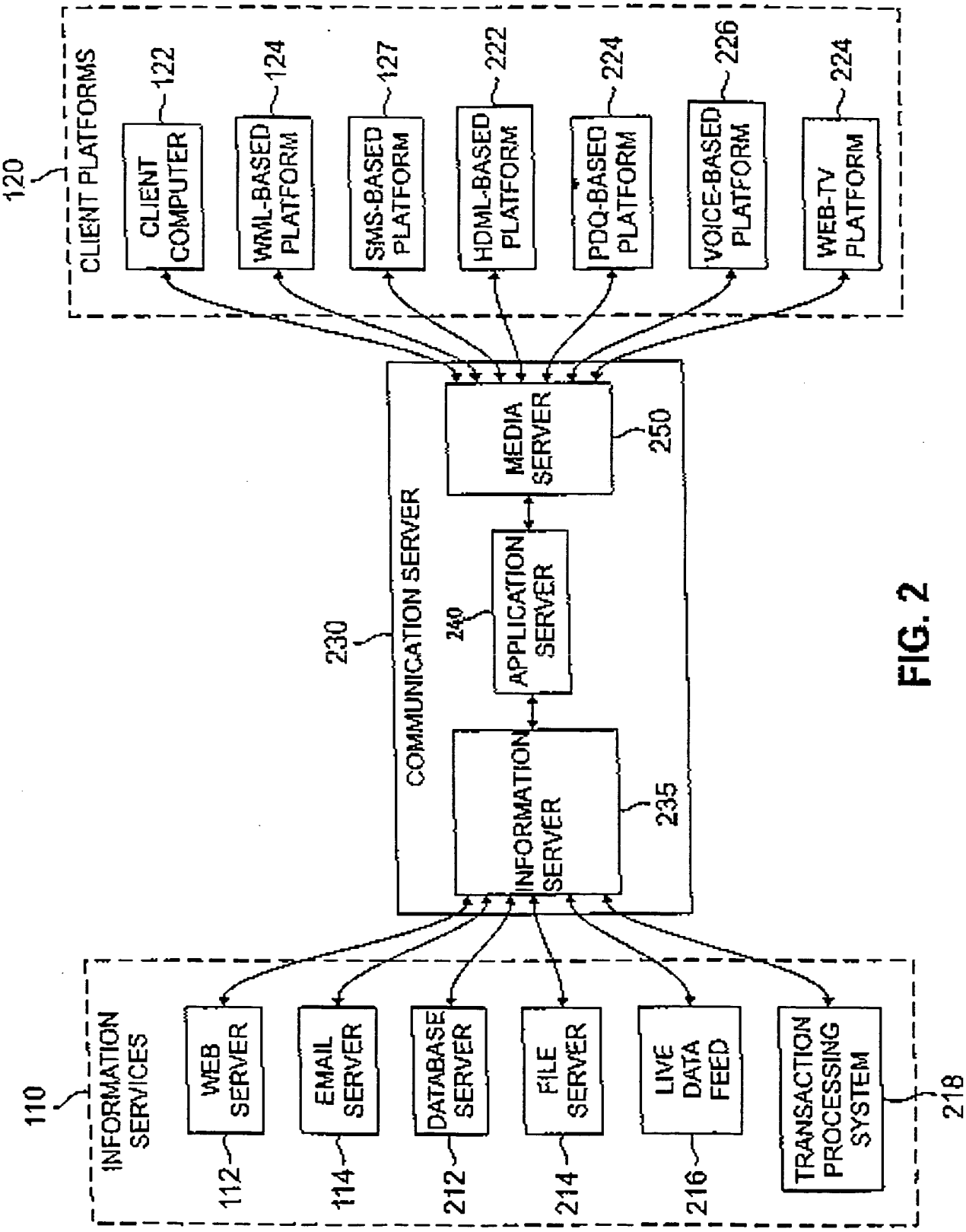

CLIENT PLATFORMS
120
CLIENT COMPUTER
122
WML-BASED PLATFORM
124
SMS-BASED PLATFORM
127
HDML-BASED PLATFORM
222
PDQ-BASED PLATFORM
224
VOICE-BASED PLATFORM
226
WEB-TV PLATFORM
224
COMMUNICATION SERVER
230
MEDIA SERVER
250
APPLICATION SERVER
240
INFORMATION SERVER
235
INFORMATION SERVICES
110
WEB SERVER
112
EMAIL SERVER
114
DATABASE SERVER
212
FILE SERVER
214
LIVE DATA FEED
216
TRANSACTION PROCESSING SYSTEM
218

FIG. 2

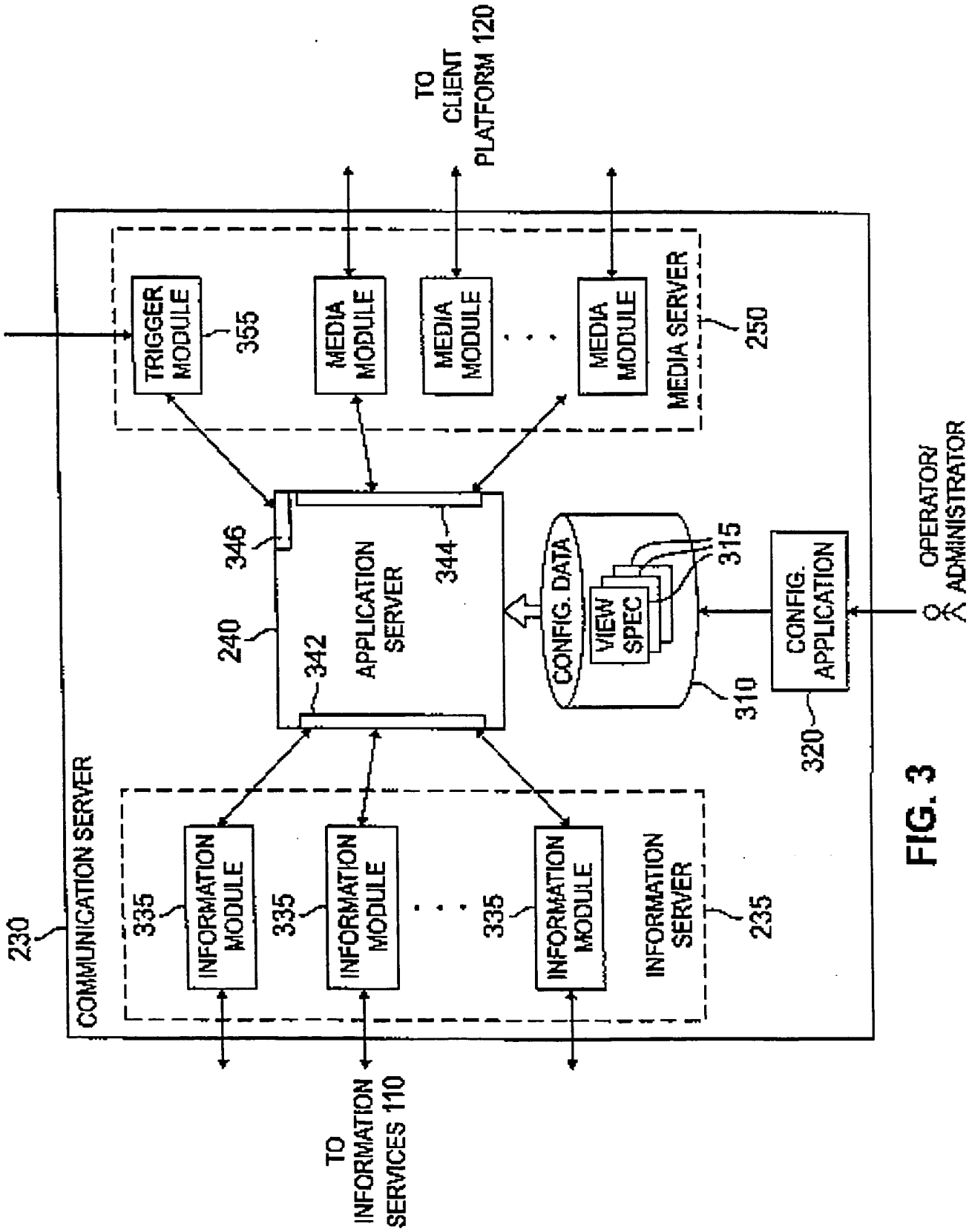

TO CLIENT PLATFORM 120
TRIGGER MODULE
355
MEDIA MODULE
MEDIA MODULE
MEDIA MODULE
MEDIA SERVER
250
346
344
APPLICATION SERVER
240
342
CONFIG. DATA
VIEW SPEC
315
310
CONFIG. APPLICATION
320
OPERATOR/ ADMINISTRATOR
230
COMMUNICATION SERVER
335
INFORMATION MODULE
335
INFORMATION MODULE
335
INFORMATION MODULE
INFORMATION SERVER
235
TO INFORMATION SERVICES 110

FIG. 3

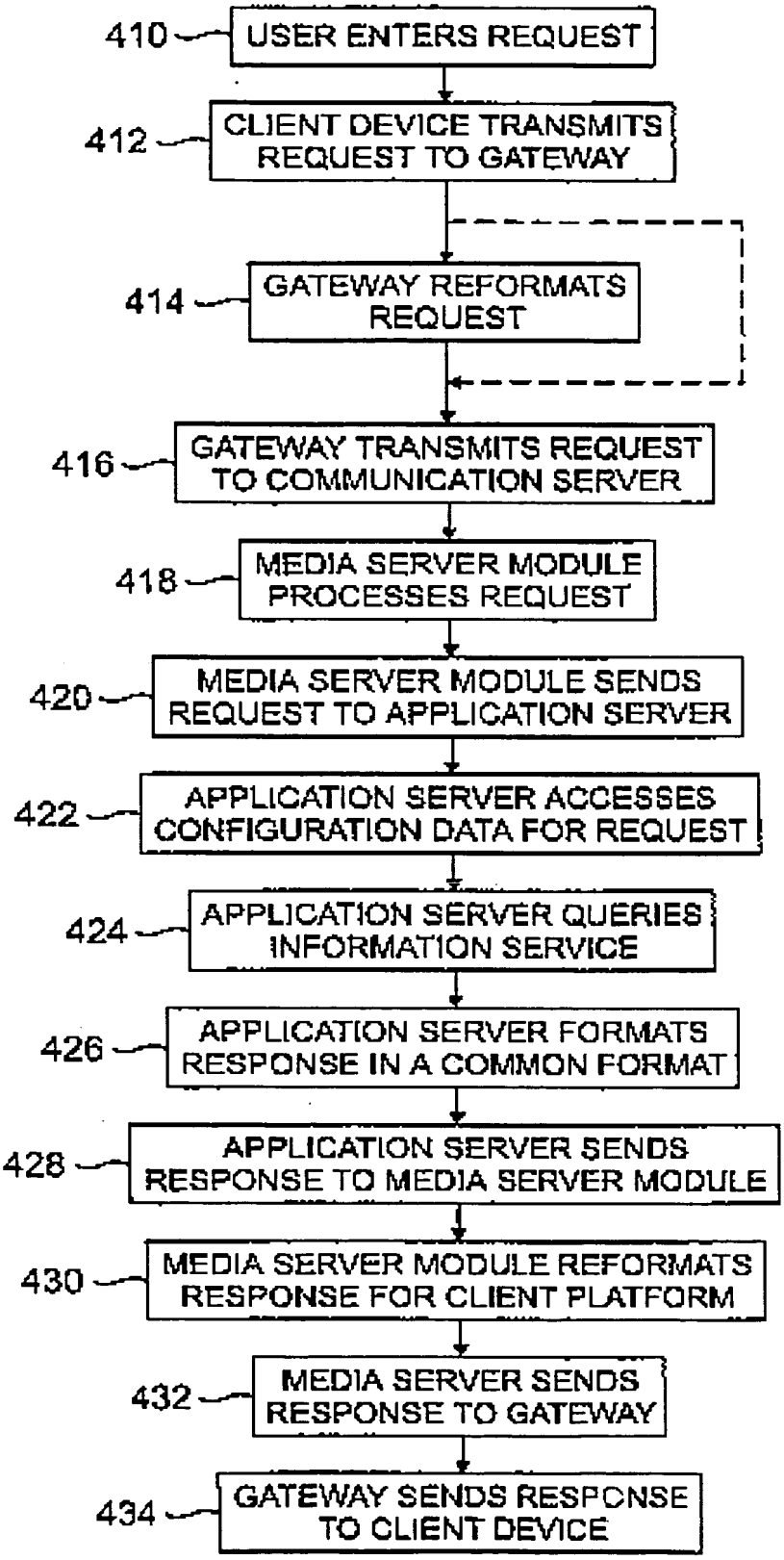

410
USER ENTERS REQUEST
412
CLIENT DEVICE TRANSMITS REQUEST TO GATEWAY
414
GATEWAY REFORMATS REQUEST
416
GATEWAY TRANSMITS REQUEST TO COMMUNICATION SERVER
418
MEDIA SERVER MODULE PROCESSES REQUEST
420
MEDIA SERVER MODULE SENDS REQUEST TO APPLICATION SERVER
422
APPLICATION SERVER ACCESSES CONFIGURATION DATA FOR REQUEST
424
APPLICATION SERVER QUERIES INFORMATION SERVICE
426
APPLICATION SERVER FORMATS RESPONSE IN A COMMON FORMAT
428
APPLICATION SERVER SENDS RESPONSE TO MEDIA SERVER MODULE
430
MEDIA SERVER MODULE REFORMATS RESPONSE FOR CLIENT PLATFORM
432
MEDIA SERVER SENDS RESPONSE TO GATEWAY
434
GATEWAY SENDS RESPONSE TO CLIENT DEVICE

FIG. 4

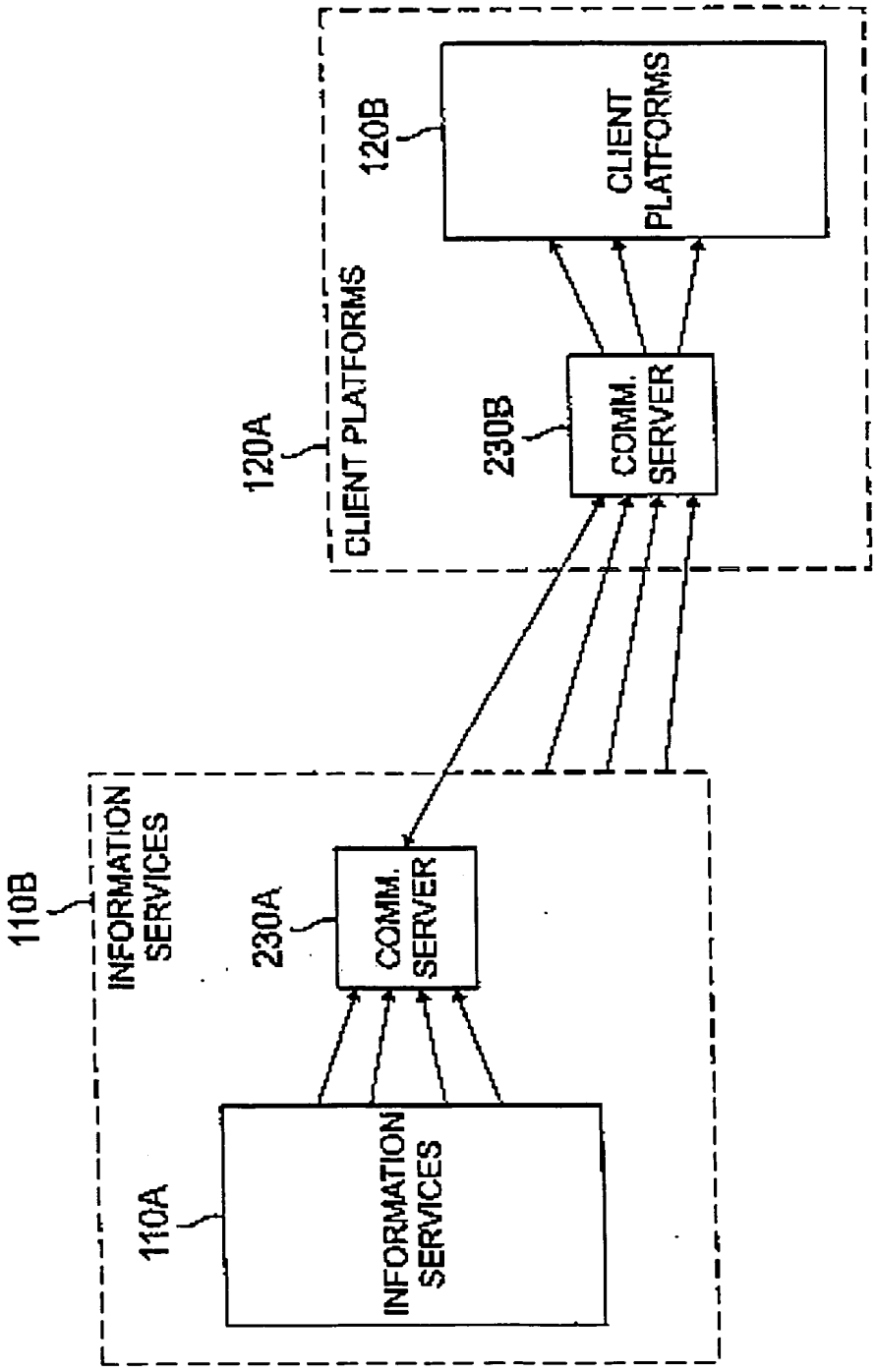

120B
CLIENT PLATFORMS
120A
CLIENT PLATFORMS
230B
COMM. SERVER
110B
INFORMATION SERVICES
230A
COMM. SERVER
110A
INFORMATION SERVICES

FIG. 5

THIS PAGE BLANK (USPTO)